DOCUMENT RESUME

ED 118 396 SE 020 210

AUTHOR Larson, Allen L.
TITLE Bridging the Gap from Gates to Microprocessors in Ten Hours.
PUB DATE Jun 75
NOTE 11p.; Paper presented at the Annual Meeting of the American Society for Engineering Education (Colorado State University, Ft. Collins, Colorado, June 16-19, 1975)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage
DESCRIPTORS College Science; *Computer Assisted Instruction; Curriculum Development; *Engineering Education; Higher Education; Instruction; *Programed Instruction; Science Education; *Teaching Methods; *Technology
IDENTIFIERS *Educomp

ABSTRACT

The advent of microprocessor technology requires that the teaching of computer basics become a part of undergraduate digital electronics courses. This paper describes a laboratory-oriented approach to basic instruction requiring less than 10 hours student study time. The technique utilizes a programmed text and a small 4-bit computer, "Edcomp," designed for the purpose of teaching computer fundamentals. The approach taken, the design of "Educomp," and the topics covered in the programmed text are presented. (Author/CP)

BRIDGING THE GAP FROM GATES TO MICROPROCESSORS IN TEN HOURS[1]

Allen L. Larson, PhD

Abstract:

Microprocessor technology requires that the teaching of computer basics become a part of undergraduate digital electronics courses. A "hands on" approach to covering these basics is described. The approach utilizes a low-cost TTL 4-bit modular computer and programmed text to cover the material in less than ten hours student study time.

1. Originally presented at the Annual Conference of American Society for Engineering Education, 16-19 June 1975 at Colorado State University, Fort Collins, Colorado.

I. INTRODUCTION

Microprocessor technology has created a problem for electrical engineering educators: The computer is now a universal circuit component, and cannot be ignored in digital circuits courses. Further, the usefulness and low cost of microprocessors has caused a host of non-EE majors to seek a working knowledge of these devices. But proper use of microprocessors requires a fairly extensive background knowledge of the inner workings of digital computers. Unfortunately, most curricula no longer allow the time for a leisurely coverage of this background material, especially in service courses.

To solve this problem, a special training aid and programmed text was developed at the Air Force Academy. A small 4 bit computer, called "Educomp," was designed with the sole purpose of teaching computer fundamentals. This machine and the programmed text has been used very successfully in two courses (one EE major's course and one service course) by over 120 students, most of whom were not engineering majors. Nearly all students required less than ten hours actual work to complete the material. Enthusiasm was high, generated by the "hands on" operation and the "knowledge-per-unit-time return ratio".

This paper discusses the approach taken, the design of "Educomp", and the topics covered in the programmed text.

II. THE APPROACH

Figure 1 shows the viewpoint the author takes of a computer. A computer is many things, and can usefully be viewed as a spectrum of topics. When viewed in this manner, one sees that most students are usually introduced to computers from the software end of the spectrum. Normally, only those EE majors interested in computer design are exposed to the complete hardware portion of the spectrum.

Typically, course work in this area takes several semesters, because the student must learn the design methods and trade-offs of each topic. But what is needed first with microprocessors is a general knowledge of how a computer works, rather than this highly specialized design knowledge. Only after a student has learned these basics might he need design knowledge, and then only if he needs it for interfacing the microcomputer into his system.

Educomp, then, gives students this background information, and serves as a foundation for several possible later developments, such as:

1. A microprocessor system design course
2. A microprocessor applications course
3. A computer architecture course
4. A logic systems design course
5. An assembly language programming course[1]
6. A digital signal processing course[1]
7. A digital controls course[1]

The key to the rapid coverage of this material is Educomp's design, described in the next section.

III. THE DESIGN OF EDUCOMP

The philosophy which guided the development of Educomp is a principle attributed[2] to Hamilton[3] which can be paraphrased as "Whenever studying or developing any new material or theory, use the simplest possible example which is adequate to illustrate the basic principles involved". This resulted in the final specifications of Educomp:

1.	Word length	4 bits
2.	Instructions	16 total
3.	Memory	16 words

1. Using a micro- or minicomputer
2. By Professor T. J. Higgins, University of Wisconsin, Madison.
3. Sir William Rowan Hamilton (1805-1865) was a famous Irish mathematician who developed the "Hamiltonian" upon which most of our optimal control theory is based.

4.	I/O	1 input port
		1 output port
5.	Peripherals	A/D
		D/A
		Others using I/O Ports
6.	Registers	Single accumulator

Figure 2 shows the organization of Educomp. For ease of comprehension, several units were designed in the most forward manner possible. For example, six separate integrated circuits are used to do the NOT, ADD, SUB, AND, and OR operations on the accumulator (ACC) and memory (MEM), as required. Then, an eight-way data selector routes the appropriate output into the accumulator. Other positions of the selector use the result of direct wire connections to route the input (INP), memory (MEM) or rotated accumulator (RAR) data back into the acculator.

The packaging of Educomp is based upon the concept of maximum visability. Over seventy LED's display all data, control, and status lines. Additionally, 16 LED's display the result of instruction decoding. All inputs are performed with switches. The physical layout of the units follows the block diagram exactly, with IC'S placed within the appropriate blocks and LED's located on or near the appropriate connecting lines.

Educomp is constructed using the wire-wrapping technique. The parts are mounted on a specially prepared piece of perforated fiberglass board stock. The top of the board is covered with a sheet of double-sided adhesive, over which is laid the block diagram. Then the top side is covered with clear contact plastic sheet. The bottom is similarly covered with a sheet giving wiring information. Sockets and components are then added, and the board is wired and mounted as the top of a plastic box. The result is a low cost, durable package.

The basic computer is packaged as

1. I/O Unit
2. Memory Unit
3. Arithmetic Unit
4. Control Unit - Part 1
5. Control Unit - Part 2
6. Analog Interface Unit
7. Manual Test Unit

This modular packaging allows the text to precede in a logical manner through the material, as discussed in the next section.

IV. THE PROGRAMMED TEXT

The organization of the programmed text is presented in Table 1. The order of the material follows a logical path through the computer spectrum and has proven to be quite efficient.

One should note the emphasis on applications at the end of the course. Students do four programs in the areas of digital and analog applications. In each area, three of these programs are examples worked through, while the fourth is done entirely by the student.

The programming deserves mention. A special form, shown in part in Figure 3, combines a memory map, a programming form, and an op code table. This single form is used for assembly language programming, hand assembly, program loading, and debug.

As an example of using Educomp to build upon, one short assignment given in the EE major's course was to write an Algol program to cross-assemble Educomp programs. All students completed the assignment in a week, despite inexperience.

V. SUMMARY

This paper has presented the basic ideas behind Educomp, which uses a combination of hardware and software (text)

to cover basic computer operations and the background needed to effectively begin using microprocessors. It is hoped that other educators will find this approach useful. Construction costs per Educomp are approximately $200, exclusive of labor. Others may well develop their own Educomp's, or adopt the USAFA Educomp. Complete construction information will be made available approximately September 1975.

TABLE 1 - COVERAGE OF EDUCOMP MATERIAL.

TOPIC	UNITS REQUIRED	MATERIAL COVERED
Introduction	None	Overview, organization
Basic Logic Blocks	None	AND, OR, NOT Gates, n-way selectors, decoders, flip-flops, registers, memory
I/O Unit	I/O, M.T.[1]	Patching, registers, loading, data selectors, control signals
Memory Unit	Memory	Addressing, data storage, instruction storage, memory
Arithmetic Unit	Arithmetic, M.T.	Logic operations, addition, subtraction, bit masking, status signals, number representations, machine instructions
Control Unit-Part 1	Control 1, M.T.	Program bookkeeping, program counter, memory addressing, instruction set, fetch-execute cycle, algorithms, instruction register, instruction decoding
Manual Operation as a computer	I/O, Mem., Arith., Control 1, M.T.	State diagram, program storage and execution, manual state controller simulation, clocking
Automatic Operation and Programming	I/O, Mem, Arith, Control 1 & 2, M.T.	"One hot" state controller, input synchronization, assembly language programming, system operation, documentation
Digital Applications	Same as above	Loader, digital logic, parallel-serial converter, remote controller programs
Analog Applications	As above, plus Analog	A/D, D/A, real-time programming, rectifier, voltage divider, voltage clamper programs

1. M.T.=Manual Test Unit

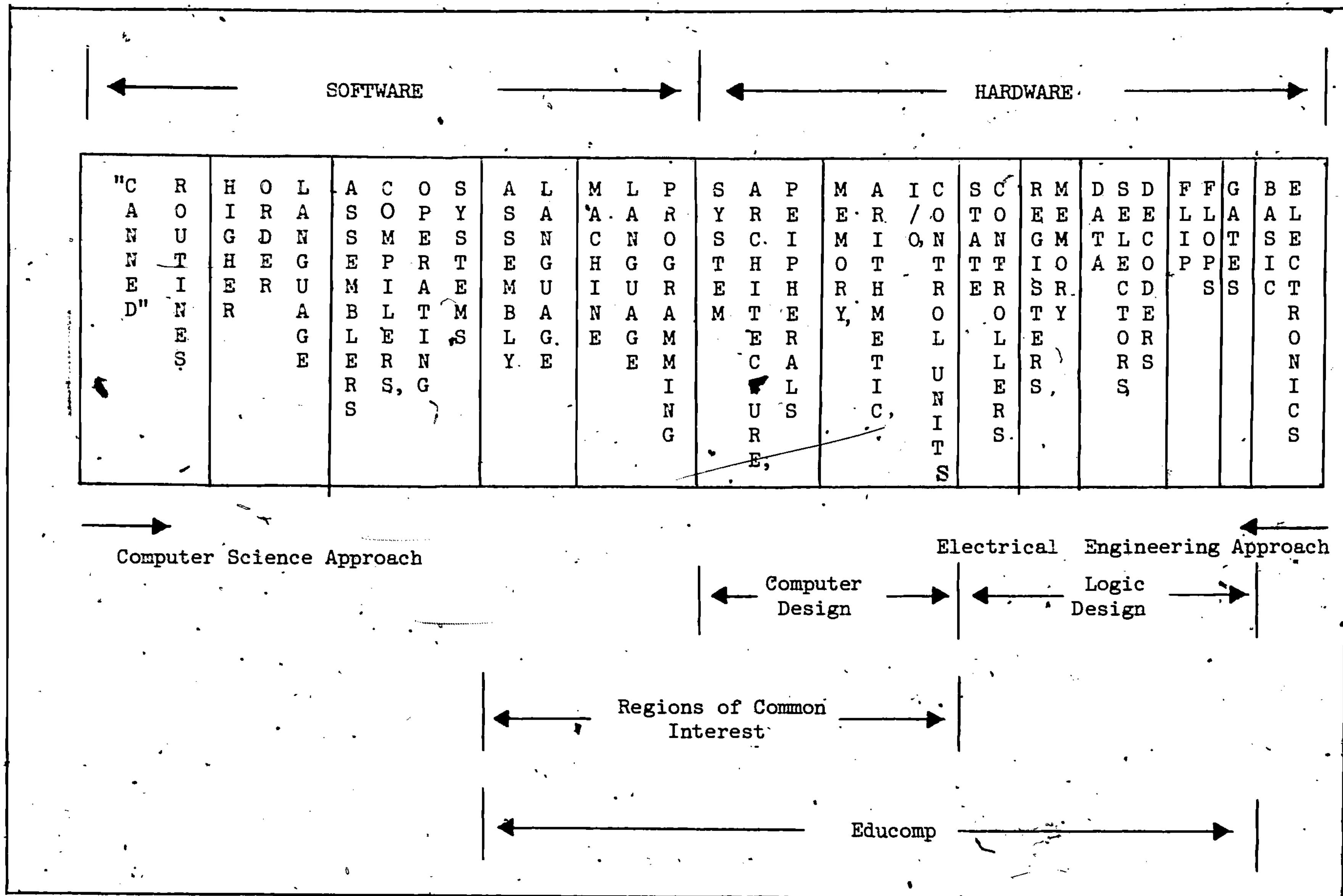

Figure 1. The Computer As A Spectrum

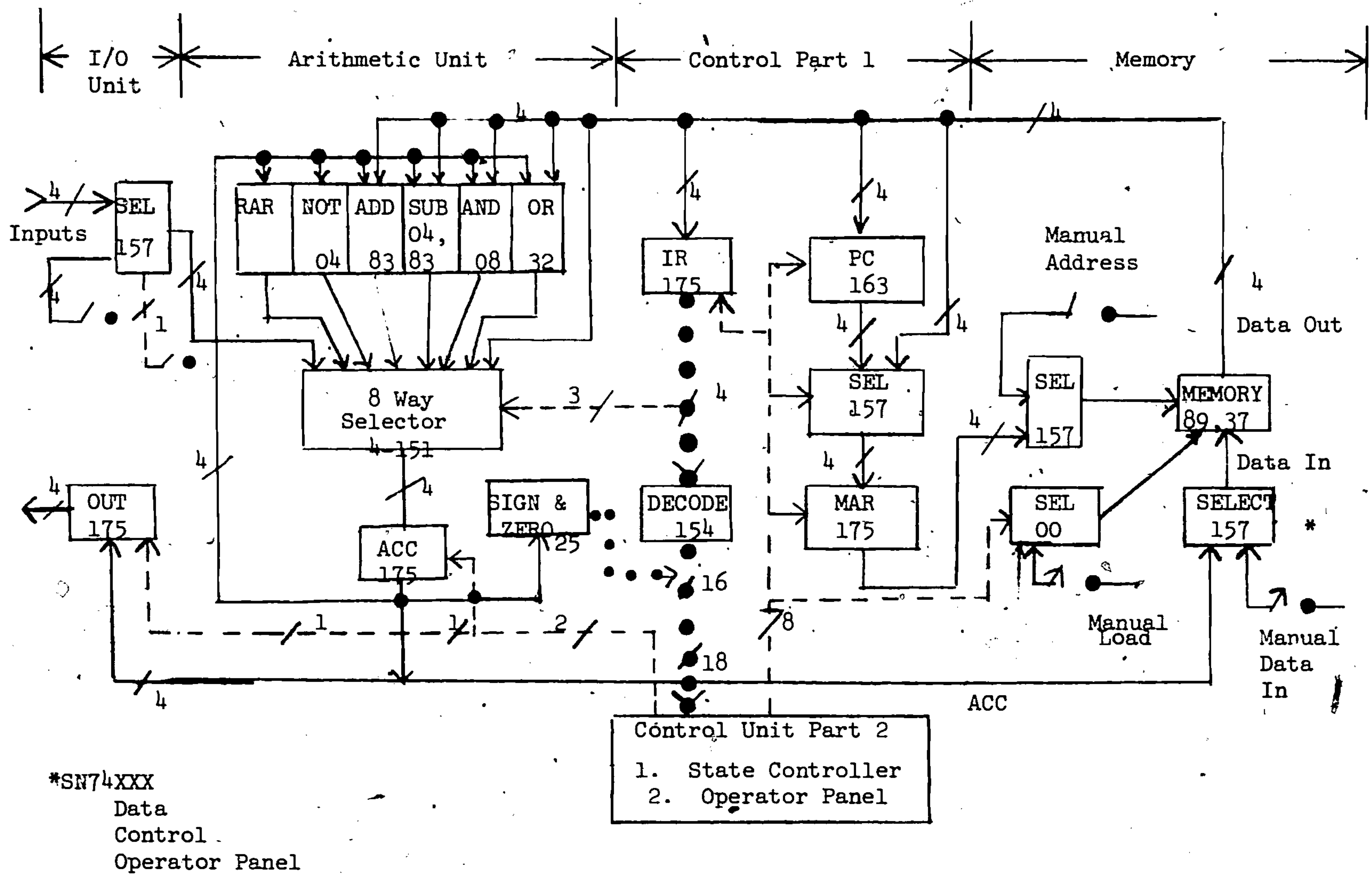

Figure 2. Educomp Architecture

EDUCOMP PROGRAMMING FORM

MEMORY ADDRESS	CONTENTS	LABEL	OP CODE	OPERAND	COMMENTS
0000		START:	INP		Get First #
0001			STA	NUM1	Store In
0010					Memory
0011			INP		Get Second #
0100			ADD	NUM1	Add to
0101					First Number
0110			JMP	START	Repeat
0111					
1110					
1111		NUM1:			Storage location for first #

INP=0000	HLT=0100	JMP=1000	SUB=1100
RAR=0001	NOP=0101	JZ =1001	AND=1101
NOT=0010	RST=0110	JP =1010	OR =1110
OUT=0011	STA=0111	ADD=1011	LDA=1111

ONE BYTE ← → TWO BYTES

Figure 3. Educomp Programming Form